Lang Resources & Evaluation (2013) 47:1007–1029
DOI 10.1007/s10579-013-9215-6

ORIGINAL PAPER

GATE Teamware: a web-based, collaborative text annotation framework

Kalina Bontcheva · Hamish Cunningham · Ian Roberts · Angus Roberts · Valentin Tablan · Niraj Aswani · Genevieve Gorrell

Published online: 2 February 2013

Abstract This paper presents GATE Teamware—an open-source, web-based, collaborative text annotation framework. It enables users to carry out complex corpus annotation projects, involving distributed annotator teams. Different user roles are provided (annotator, manager, administrator) with customisable user interface functionalities, in order to support the complex workflows and user interactions that occur in corpus annotation projects. Documents may be pre-processed automatically, so that human annotators can begin with text that has already been pre-annotated and thus making them more efficient. The user interface is simple to learn, aimed at non-experts, and runs in an ordinary web browser, without need of additional software installation. GATE Teamware has been evaluated through the creation of several gold standard corpora and internal projects, as well as through external evaluation in commercial and EU text annotation projects. It is

K. Bontcheva (✉) · H. Cunningham · I. Roberts · A. Roberts · V. Tablan · N. Aswani · G. Gorrell
Department of Computer Science, University of Sheffield, Regent Court, 211 Portobello, Sheffield S1 4DP, UK
e-mail: K.Bontcheva@dcs.shef.ac.uk

H. Cunningham
e-mail: H.Cunningham@dcs.shef.ac.uk

I. Roberts
e-mail: I.Roberts@dcs.shef.ac.uk

A. Roberts
e-mail: A.Roberts@dcs.shef.ac.uk

V. Tablan
e-mail: V.Tablan@dcs.shef.ac.uk

N. Aswani
e-mail: N.Aswani@dcs.shef.ac.uk

G. Gorrell
e-mail: G.Gorrell@dcs.shef.ac.uk

available as on-demand service on GateCloud.net, as well as open-source for self-installation.

Keywords Text annotation · Web-based annotation tool · GATE · Cloud-based text annotation service

1 Introduction

For the past ten years, Natural Language Processing (NLP) frameworks such as GATE (Cunningham et al. 2011b) and UIMA (Ferrucci and Lally 2004) have been providing tool support and facilitating NLP researchers with the task of implementing new algorithms, sharing, and reusing them. At the same time, NLP research was driven forward by a growing volume of annotated text corpora, produced by projects and evaluation initiatives such as ACE (2004), TAC,[1] SemEval and Senseval (www.senseval.org). Some NLP frameworks (e.g. AGTK Maeda and Strassel 2004); GATE (Cunningham et al. 2002; Cunningham et al. 2011b) also provide text annotation user interfaces. For instance, GATE has been used to create the MPQA corpus (Wiebe et al. 2005) and also in the American National Corpus project (Ide and Sudrman 2005).

However, much more is needed in order to produce high quality annotated text corpora: a stringent methodology, annotation guidelines, inter-annotator agreement (IAA) measures, and often, annotation adjudication (or data curation) to reconcile differences between annotators. All these make the text annotation process expensive and complex to manage (Hovy 2010).

We argue that corpus annotation can be made significantly simpler, through a multi-role methodological framework to support the different phases and actors in the text annotation process. The multi-role support is particularly important, as it enables the most efficient use of the skills of the different people. It also lowers overall annotation costs, through simple and efficient annotation web-based UIs for non-specialist annotators. This also enables role-based security, project management and performance measurement of annotators, which are all particularly important when creating corpora with relatively unskilled annotators. Having a multi-stage, multi-role annotation process has also been shown to improve annotation quality and speed [e.g. OntoNotes (Hovy et al. 2006)].

This paper presents GATE Teamware,[2] an open-source text annotation framework and a methodology for the implementation and support of complex annotation projects. It has a web-based architecture, where a number of web services (e.g. document storage, automatic annotation) are made available via HTTPS and the users interact with the text annotation interfaces through a standard web browser.

GATE Teamware is based on GATE (Cunningham et al. 2011b), a widely used, scalable and robust open-source NLP platform. GATE comes with numerous reusable text processing components for many natural languages, coupled with a

[1] http://www.nist.gov/tac/.

[2] Source code and documentation are available from http://gate.ac.uk/teamware/.

graphical NLP development environment and user interfaces for visualisation and editing of linguistic annotations, parse trees, co-reference chains, and ontologies. GATE Teamware however was created specifically to be used by non-expert annotators, as well as to enable methodologically sound, efficient, and cost-effective corpus annotation projects over the web.

In addition to its research uses, GATE Teamware has also been tested as a framework for cost-effective commercial annotation services, supplied either as in-house units or as outsourced specialist activities. Several test annotation projects have been conducted in the domains of bio-informatics and business intelligence, with minimal training and producing high quality corpora. For example, Meurs et al. (2011) apply GATE Teamware to the task of building a database of fungal enzymes for biofuel research. Their results show that using GATE Teamware for automatic pre-annotation and manual correction increases the speed with which papers can be processed for inclusion in the database by a factor of around 50 %.

Similar to other server-side software, GATE Teamware installation is a specialised, non-trivial task with associated costs, in terms of significant time and staff expertise required. In order to lower this barrier and provide zero startup costs, we have made available cloud-based GATE Teamware virtual machines,[3] that can be turned on and off as required. In addition, the GATECloud.net (Tablan et al. 2013) integration makes it easy to choose a set of automatically annotated documents and send these into a GATE Teamware instance. There is also a virtual machine distribution that can be downloaded and run locally instead.

The rest of the paper is structured as follows. Section 2 defines the requirements which need to be met by web-based collaborative annotation tools in general. It is followed by Sect. 3, which discusses related work, in the context of these requirements. The GATE Teamware architecture and implementation are detailed in Sect. 4. Section 5 presents usage evaluation results, followed by Sect. 6, which discusses the cloud-based GATE Teamware service, available on-demand.

2 Requirements for multi-role, web-based collaborative annotation environments

As discussed above, collaborative text annotation is a complex process, which involves different kinds of actors and requires a wide range of automatic pre-processing, user interface, and evaluation tools. From a high-level methodological perspective, web-based text annotation frameworks need to support annotation efficiency, consistency, scale, good interfaces, and clear procedures (Hovy 2010).

These translate into a set of functional requirements, which need to be met:

1. *Multi-role support*, including user groups, access privileges, annotator training, quality control, and corresponding user interfaces.
2. *Shared, efficient data storage* to store and access text corpora and annotations.
3. *Support for automatic pre-annotation services* and their configuration, to help achieve time and cost savings.

[3] Available to use and trial at http://gatecloud.net.

4. *Flexible workflow engine* to model complex annotation methodologies (e.g. Hovy 2010) and interactions.
5. *Web-based user interfaces*, that include customisable templates for common annotation tasks and support annotator comments.
6. *Support for relevant linguistic annotation standards*, especially ISO/TC 37/SC 4 (Ide and Romary 2004).

Next we discuss the first four functional requirements in further detail.

2.1 Multi-role support and division of labour

Due to annotation projects having different sizes and complexity, in some cases the same person might perform more than one role or new roles might be needed. For example, the person managing the project might also sometimes be an annotator. Methodologically speaking, it is also possible to distinguish between adjudicators (the people who reconsile disagreements between annotators) and managers, who setup the project, provide annotation guidelines, etc. However, following some user feedback, GATE Teamware currently has merged these two roles into one—manager. The administrator interface does support the definition of new roles and the corresponding configuration of the web user interface components, so it would be possible to separate these roles in the future, if the need arises.

In more detail, we argue that it is necessary to distinguish the following three user roles.

Annotators are given a set of annotation guidelines and often work on the same document independently and concurrently. In order to enable less-specialised annotators to be used, manual annotation user interfaces need to be simple to learn. In addition, there needs to be an automatic training mode for annotators where their performance is compared against a known gold standard and all mistakes are identified and explained to the annotators, until they have mastered the guidelines.

Since annotators and project managers are often working at different locations, there needs to be a communication channel between them, e.g. instant messaging. If a manager is not available, an annotator should also be able to mark an annotation as requiring discussion and then all such annotations should be shown automatically in the manager console. Annotators need to be able to save their work and, if they close the annotation tool, the same document must be presented to them for completion next time they log in. Optionally, some projects might wish to restrict annotators to working on a maximum of n documents (given as a number or percentage), in order to prevent an over-zealous annotator from taking over a project and potentially introducing individual bias.

From an user interface perspective, there needs to be support for annotating document level metadata (e.g. language identification), word-level annotations (e.g. named entities, POS tags), and relations and trees (e.g. co-reference, syntax trees). Ideally, the interface should offer some generic components for all these, which can be customised with project-specific tags and values via an XML schema or other similar declarative mechanisms. The framework also needs to be extensible, so specialised UIs can easily be plugged in, if required.

Project managers are typically in charge of defining new corpus annotation projects and their workflows, monitoring annotation progress, dealing with annotator performance issues, and carrying out annotator training. They also define the annotation guidelines, the associated schemas (or set of tags), and prepare and upload the corpus to be annotated. Managers also make methodological choices: whether to have multiple annotators per document; how many; which automatic NLP services need to be used to pre-process the data; and what is the overall workflow of annotation, quality assurance, adjudication, and corpus delivery.

Managers need a project monitoring tool where they can see:

- Whether a corpus is currently assigned to a project or, what annotation projects have been run on the corpus with links to these projects or their archive reports (if no longer active). Also links to the the annotation schemas for all annotation types currently in the corpus.
- Project completion status (e.g., 80 % manually annotated, 20 % adjudicated).
- Annotator statistics within and across projects: which annotator worked on which document, how long they took, and what was their IAA (if measured).
- The ability to lock a corpus from further editing, either during or after a project.

Finally, managers are responsible for annotation adjudication and gold-standard production. Therefore, in addition to the standard annotation interfaces, they have access to the IAA user interface (appropriate for comparing the differences between two annotators only). They also need a specialised adjudication interface which helps them identify and reconcile differences in multiply annotated documents. Even though manual curation adds to the cost of corpus annotation, it is typically very beneficial to include that as part of the workflow, since it improves the annotation quality in hard-to-solve cases (Hovy 2010).

Administrators define roles for other users, create user accounts, create and configure services, and monitor workflow processes.

2.2 Remote, scalable data storage

Given the multiple user roles and the fact that several annotation projects may be running at the same time with different remotely located teams, the data storage layer needs to scale to accommodate large, distributed corpora and have the necessary security in place through authentication and fine-grained user/group access control (Brugman et al. 2004).

Data security is paramount and needs to be enforced as data is being sent over the web to the remote annotators. Support for diverse document input and output formats is also necessary, especially stand-off ones (e.g. XCES Ide et al. 2000) which can minimise network traffic by transmitting only a relevant subset of all annotations.

Since multiple users must be able to work concurrently on the same document, there needs to be an appropriate locking mechanism to support that. The data storage layer also needs to provide facilities for storing annotation guidelines, annotation schemas, and, if applicable, ontologies. Last, but not least, a corpus

search functionality is often required, at least one based on keywords, but ideally also including document metadata and linguistic annotations.

2.3 Automatic pre-annotation services

Automatic pre-annotation services can reduce significantly annotation costs (e.g. annotation of named entities), but unfortunately they also tend to be domain or application specific. Also, several might be needed in order to bootstrap all annotation types, e.g. named entities, co-reference, and relation annotation modules. Therefore, the architecture needs to be open so that new services can be added easily. Such services can encapsulate different NLP modules and take as input one or more documents (or an entire corpus). The automatic services also need to be scalable, in order to minimise their impact on the overall project completion time. The project manager should also be able to choose services based on their accuracy on a given corpus.

Machine Learning (ML) IE modules can be regarded as a specific kind of automatic service. A mixed initiative system (Day et al. 1997) can be set up by the project manager and used to facilitate manual annotation behind the scenes. This means that once a document has been annotated manually, it will be sent to train the ML service which internally generates an ML model. This model will then be applied by the service on any new document, so that this document will be partially pre-annotated. The human annotator then only needs to validate or correct the annotations provided by the ML system, which makes the annotation task significantly faster (Day et al. 1997).

2.4 Flexible workflow engine

In order to have an open, flexible model of corpus annotation processes, we need a powerful workflow engine which supports asynchronous execution and arbitrary mix of automatic and manual steps. For example, manual annotation and adjudication tasks are asynchronous. Resilience to failure is essential and workflows need to save intermediary results from time to time, especially after operations that are very expensive to re-run (e.g. manual annotation, adjudication). The workflow engine also needs to have status persistence, action logging, and activity monitoring, which form the basis of the project management tools.

In a workflow it should be possible for more than one annotator to work on the same document at the same time; however, during adjudication, all affected annotations need to be locked to prevent concurrent modifications. For separation of concerns, it might be useful for the same corpus to have more than one active projects. Similarly, the same annotator needs to be able to work on several annotation projects.

3 Related work

There are a number of pre-existing tools for corpus annotation, both for textual and multimedia corpora. Table 1 provides a high-level overview against the six

Table 1 Nine different annotation tools listed against their ability to meet the six different requirements

Tool	Multi-role support	Shared data storage	Pre-annot. services	Annotation workflows	Web UI	ISO TC37/SC4 compliance
Callisto	No	No	No	No	No	No
MMAX2	IAA only	No	No	No	No	No
GATE	IAA only	No	Yes	No	No	Yes
Knowtator	Some	No	No	No	No	No
NXT	No	No	Yes	No	No	Planned
ANNEX	No	Yes	No	No	View only	Yes
Atlas.ti	Some	Yes	No	No	Yes	No
LDC tools	Partial	Yes	Yes	No	Yes	No
OntoNotes	Partial	Yes	Yes	No	Yes	No

requirements identified above. The rest of this section will discuss them in more detail.

3.1 Stand-alone annotation tools

Callisto (Day et al. 2004) is a stand-alone linguistic annotation workbench, designed specifically to be easily extendable with task-specific annotation interfaces, e.g. named entities, relations, time expressions. MMAX2 (Müller and Strube 2006) is another stand-alone text annotation tool, which uses stand-off XML and annotation schemas for customisation. MMAX2 also computes IAA and provides a query language for searching over the corpus, as well as an API for programmatic access to the annotation objects. Similar functionalities are offered by GATE Developer (Cunningham et al. 2011b), augmented with support for XCES (Ide et al. 2000) and the ability to pre-annotate corpora automatically.

A somewhat more complex approach is to model the different linguistic annotation tasks with ontologies, which is the approach taken by Knowtator (Ogren 2006). In addition to measuring IAA, Knowtator also supports semi-automatic adjudication and the creation of a consensus annotation set.

In the area of multi-modal corpus annotation, the NITE XML toolkit (NXT) (Carletta et al. 2005) is a stand-alone annotation tool, which puts special emphasis on the editing and querying of complex, cross-annotated corpora. Another similar desktop-based tool is ELAN (Brugman and Russel 2004). Early experiments on adapting ELAN for collaborative annotation of multimedia corpora over the web were based on a peer-to-peer architecture (Brugman et al. 2004). A more recent, web-based tool called ANNEX (Berck and Russel 2006) has been implemented; however it is limited to only viewing the multimedia annotations created by ELAN.

Even though none of these stand-alone annotation tools meet more than three of our six requirements, they have informed important design decisions in GATE Teamware. The first one is the adoption of stand-off XML, XCES, and related ISO TC37/SC4 standards to facilitate distributed editing and annotation format compliance. The second is the adoption of annotation schemas as declarative means for customising and generalising the text annotation interfaces. Thirdly, to minimise the required implementation effort, we chose to reuse the text annotation interfaces, the ANNIC corpus query and search, and the IAA plugins of GATE Developer (Cunningham et al. 2011b).

3.2 Web-based text annotation tools

The second category of related work includes a number of web-based text annotation tools. One of them is ATLAS.ti,[4] which has been developed to assist qualitative data analysis and is widely used in political science. However, it currently lacks IAA metrics, does not offer automatic pre-annotation, nor does it allow users to work simultaneously on the same document.

[4] http://www.atlasti.com/.

Some of the most sophisticated collaborative annotation tools are those developed by the Linguistic Data Consortium, due to their need to run large-scale projects. The AGTK toolkit (Maeda and Strassel 2004) provides a shared relational database model for storing and accessing corpora on a shared server, as well as being a framework for development of collaborative annotation tools based on these shared corpora. One example is the specialised ACE annotation tool, which also comes with an accompanying tool for annotation adjudication. Maeda et al. (2008) describe ACK (Annotation Collection Kit) which is web based and uses comma-separated CSV files to define the questions which an annotator has to answer (e.g., what are the possible parts of speech of this word). In the context of machine translation, they also discuss a workflow system for post-editing machine translation results which supports different user roles (editors in this case) and the communication between them. While the LDC tool set is very impressive, the various annotation tasks are covered by separate, independent tools and in some cases, these tools are specific to a particular annotation project (e.g., ACE, GALE). Although such specialised annotation tools offer time and cost savings through annotator efficiency, they come with the overhead of needing separate installations, customisation, and support, which makes them harder to reuse in smaller corpus annotation projects.

Anaphoric Bank (Poesio et al. 2012) is a conglomeration of anaphorically annotated corpora produced by different parties using compatible formats. To facilitate the addition of further data to the corpus, a tool has been created, called Serengeti, provided over the web, thus offering the advantages of centrally administered software, not requiring users to manage installations and allowing changes to be made as required. XML schemas define the format of annotations. In addition, the "Phrase Detectives" game produces anaphoric annotations as a side-effect and allows annotations to be collected from the general population. Thus far, however, the focus is on serving the specific needs of the Anaphoric Bank project, rather than offering a comprehensive, web-based text annotation framework.

Similarly, the large, multi-site OntoNotes project (Hovy 2010) has created a number of specialised tools for managing annotations, including the STAMP textual interface, server for data storage, interfaces for word sense annotations, annotation reservation interface, and statistics module. What is lacking is a flexible annotation workflow engine that binds these together and minimises manual interventions. In addition, it is not clear what is the overhead of installing and customising these tools.

To summarise, as can be seen from Table 1, there is no single web-based collaborative annotation framework, which meets fully all six requirements defined in Sect. 2. What is still missing is a generic, web-based tool, which:

- models the roles of the different actors involved in corpus annotation projects and supports their interactions in an unified environment;
- provides a set of general purpose text annotation tools, tailored to the different user roles, e.g. a management tool with inter-annotator agreement metrics and adjudication facilities and a document annotation interface for the annotators;
- supports complex annotation workflows and provides a management console with project statistics, such as time spent per document by each of the annotators, percentage of completed documents, etc.;

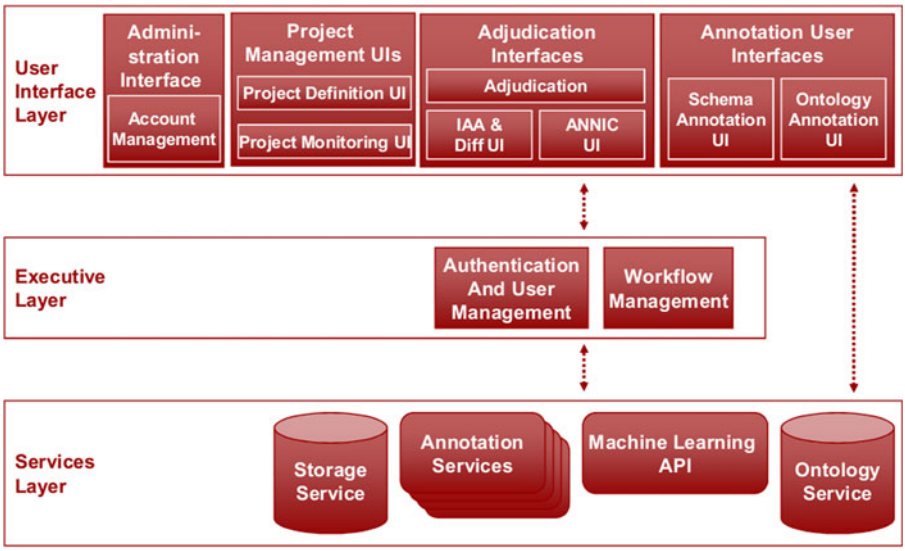

Fig. 1 GATE Teamware architecture diagram showing three layers: the user interface layer, the executive layer and the services layer

– offers built-in methodological support, to complement the tool support;
– is configurable, extensible, and compliant with relevant text annotation standards.

4 GATE Teamware: a web-based collaborative annotation and curation environment

This section introduces GATE Teamware, which is an open-source, web-based collaborative text annotation and curation environment, designed to meet all six key requirements. It supports the training and involvement of unskilled annotators, which can lower the overall cost of corpus annotation projects. Further cost reductions can be achieved through automatic pre-annotation services, if these exist for the target domain and language.

GATE Teamware is based on the GATE framework (Cunningham et al. 2011a), which provides selected user interface components, reusable automatic text annotation components, and support for linguistic annotation standards.

GATE Teamware's novelty is in being a generic, reusable, web-based framework for collaborative text annotation. Unlike other tools (see Sect. 3), GATE Teamware provides the required multi-role methodological support, as well as the necessary tools to enable the successful management of distributed annotation projects. It has a service-based architecture which is parallel, distributed, and also scalable (via service replication) (see Fig. 1). Each section of the architecture diagram will be explained in more detail below, from the bottom up.

4.1 Services layer

The services layer includes the GATE document service, serving the data structures used in GATE Teamware and the GATE annotation services, coordinating the computational tasks. Each is discussed in detail below.

4.1.1 GATE document service

The document storage service provides a distributed data store for corpora, documents, and annotation schemas. Input documents can be in all major formats (e.g., XML, HTML, PDF, ZIP), based on GATE's comprehensive support. In all cases, when a document is uploaded in GATE Teamware, the format is analysed and converted into a single unified, graph-based model of *annotation*: the one of the GATE NLP framework. Then this internal annotation format is used for data exchange between the service layer, the executive layer and the UI layer. The main export format for annotations is currently stand-off XML, including XCES (Ide et al. 2000). Multilinguality is supported via Unicode and other Java-supported text encodings.

Since some corpus annotation tasks require ontologies, these are made available from a dedicated ontology service. This wraps the OWLIM (Kiryakov 2006) semantic repository, which is needed for reasoning support and consequently justifies having a specialised ontology service, instead of storing ontologies together with documents and schemas.

4.1.2 GATE annotation services

GATE Annotation Services (GAS) provide distribution of compute-intensive NLP tasks over multiple processors. It is transparent to the external user how many machines are actually used to execute a particular service. GAS provides a straightforward mechanism for running applications, created with the GATE framework, as web services that carry out various NLP tasks. In practical applications we have tested a wide range of services such as named entity recognition (based on the freely-available ANNIE system Cunningham et al. 2002), ontology population (Maynard et al. 2009), patent processing (Agatonovic et al. 2008), and automatic adjudication of multiple annotation layers in corpora.

The GAS architecture utilises two types of components: the web service endpoint, that accepts requests from clients and queues them for processing; and one or more workers that take the queued requests and process them.

The two sides communicate using the Java Messaging System (JMS),[5] a framework for reliable messaging between Java components. If a particular service is heavily loaded it is a simple matter to add extra worker nodes to spread the load, and workers can be added or removed dynamically without needing to shut down the web services. The configuration and wiring together of these components is handled using the Spring Framework.[6]

[5] http://java.sun.com/products/jms/.

[6] http://www.springsource.org/.

Annotation pipelines, installed in GATE Teamware as a GAS, are used in projects to prepare data. GATE Teamware includes a number of pre-packaged GASes to perform common functions, such as moving and copying annotations. Managers and administrators can view and edit GASes.

4.2 The executive layer

Firstly, the executive layer implements authentication and user management, including role definition and assignment. In addition, administrators can define here which UI components are made accessible to which user roles (the defaults are shown in Fig. 1).

The second major part is the workflow manager, which is based on JBoss jBPM[7] and has been developed to meet most of the requirements discussed in Sect. 2.4 above. Firstly, it provides dynamic workflow management: create, remove, update, delete (CRUD) workflow definitions, and workflow actions. Secondly, it supports business process monitoring, i.e., measures how long annotators take, how good they are at annotating, as well as reporting the overall progress and costs. Thirdly, there is a workflow execution engine which runs the actual annotation projects. As part of the execution process, the project manager selects the number of annotators per document, the annotation schemas, the set of annotators involved in the project and the corpus to be annotated.

4.3 The user interfaces

The GATE Teamware user interfaces run in a web browser and do not require prior installation. After the user logs in, the system checks their role(s) and access privileges, to determine which interface elements they are shown. Annotators only see the annotation interfaces, whereas managers see the project management and adjudication interfaces. GATE Teamware administrators have access to all user interfaces, including a dedicated administration interface.

4.3.1 Annotation user interface

Annotators carry out manual annotation, from scratch, or by correcting automatic annotation generated by the GATE processing resources. When they log into GATE Teamware, human annotators see a very simple web page with one link to their user profile data and another one to start annotating documents.

The generic schema-based annotator UI is shown in Fig. 2. The annotation editor dialog shows the annotation types (or tags/categories) valid for the current project and optionally their features (or attributes). These are generated automatically from the annotation schemas assigned to the project by its manager. Annotation schemas define the acceptable range of annotations and attributes and thus allow the user interface to be customised, in a manner similar to other tools, such as Callisto (Day et al. 2004) and MMAX2 (Müller and Strube 2006).

[7] http://www.jboss.com/products/jbpm/.

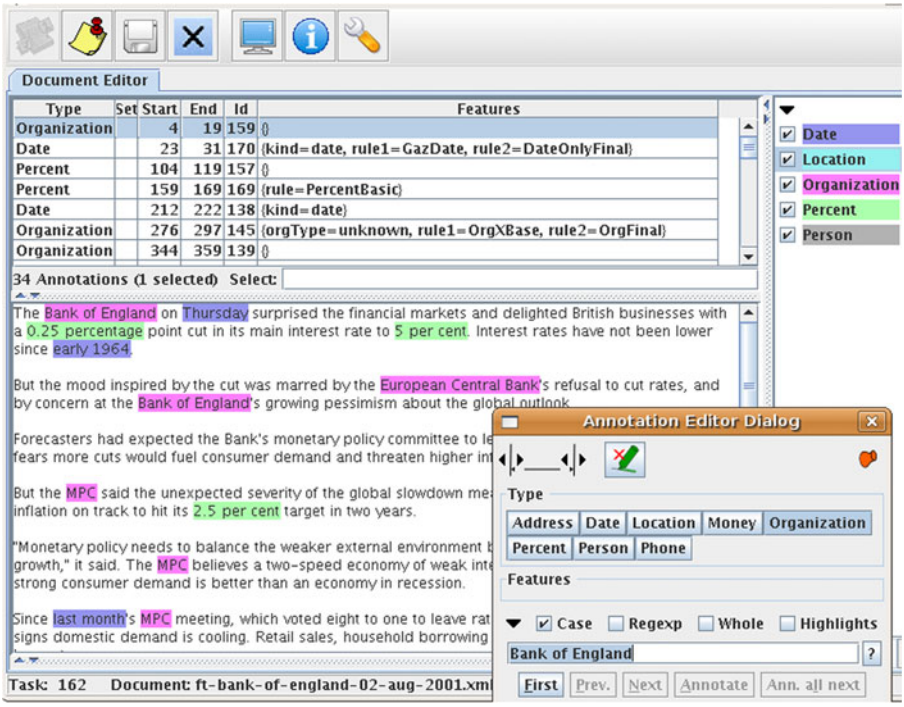

Fig. 2 The schema-based annotator user interface, showing the document displayed with annotations indicated in *coloured highlighting*, and the annotation editor dialog box, allowing annotation type to be selected from a predefined list. (Color figure online)

The annotation editor also supports the modification of annotation boundaries, either through mouse clicks or keyboard shortcuts.[8] In addition, advanced users can define regular expressions to annotate multiple matching strings simultaneously.

To add a new annotation, one selects the text with the mouse (e.g., "Bank of England") and then clicks on the desired annotation type in the dialog (e.g., Organization). Existing annotations are edited by hovering over them, which shows their current type and features in the editor dialog.

The annotation editor has a comprehensive multilingual support through Unicode—an evolution of the tools first described in (Tablan et al. 2002). Since the annotation data model underlying GATE Teamware is based on offsets, users can select any sequence of glyphs, i.e. markables are not required to be separated by white space. This is advantageous for languages, such as Thai, in which some glyphs constitute more than one Unicode character (the base character plus the tone marker). The editor also supports right-to-left, as well as left-to-right languages, through the default Java implementation.

Annotators can also control which annotation types are highlighted in the text, by selecting the corresponding check-boxes, shown at the top right side of Fig. 2. By

[8] For details see http://gate.ac.uk/userguide/sec:developer:keyboard.

default, all types are visible, but this functionality allows users to focus on one category at a time, if required.

The toolbar at the top of Fig. 2 shows all other actions which can be performed. The first button requests a new document to be annotated. When pressed, a request is sent to the workflow manager which checks if there are any pending documents which can be assigned to this annotator. The second button signals task completion, which saves the annotated document as completed on the data storage layer and enables the annotator to ask for a new one (via the first button). The third (save) button stores the document without marking it as completed in the workflow. This can be used for saving intermediary annotation results or if an annotator needs to log off before they have completed a document. The next time they log in and request a new task, they will be given this document to complete first.

Ontology-based document annotation is supported in a similar fashion, but instead of having a flat list of types on the right, the annotator is shown the type hierarchy and when they select a particular type (or class), they can then optionally choose an existing instance or add a new one.

In terms of interface design, many of the annotator interface components are reused from GATE Developer, which makes it easier for users to switch between the web-based annotation tools of GATE Teamware and the stand-alone, desktop environment of GATE Developer. In addition, this also minimises implementational effort, since the same code can be reused in both applications. The only downside of this approach is that the ergonomics of the GATE Teamware web-based annotation interface could have been more similar to other commonly used web applications, instead of using Java Web Start and Swing.

4.3.2 Adjudication interfaces

As discussed in Sect. 2.1, project managers carry out quality assurance tasks. Tools available include IAA metrics (including f-measure and Kappa) to identify if there are differences between annotators; a visual annotation comparison tool to see quickly where the differences are per annotation type; and an editor to edit and reconcile annotations manually (i.e. adjudication) or by using external automatic services.

The key part of the manual adjudication UI is shown in Fig. 3: the UI shows also the full document text above the adjudication panel, as well as lists all annotation types on the right, so the project manager can select which one they want to work on. In our example, the manager has chosen to adjudicate Date annotations created by two annotators and to store the results in a new consensus annotation set. The adjudication panel has on top arrows that allow managers to jump from one difference to the next, thus reducing the required effort. The relevant text snippet is shown and below it are shown the annotations of the two annotators. The manager can easily see the differences and correct them, e.g., by dragging the correct annotation into the consensus set. Annotation differences can also be resolved using the annotation diff interface, as shown in Fig. 4. There, the Statistics tab shows the IAA metrics, whereas the Adjudication tab (shown in focus in the figure) can be used by managers to produce the ultimate ground truth annotations.

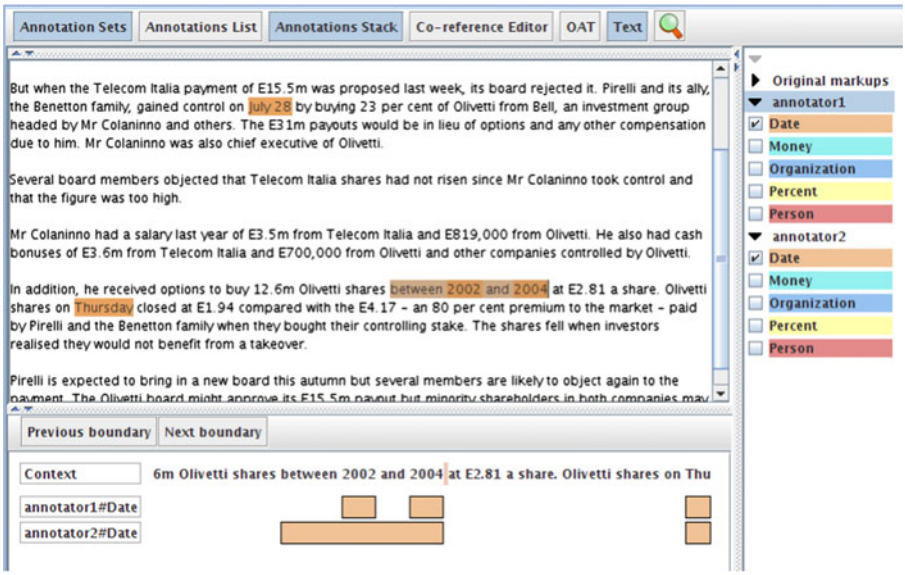

Fig. 3 Part of the adjudication user interface, showing the document displayed with *highlighted annotations* and annotation stack and list displays to the *bottom* and *right*, allowing different annotators' work to be compared by the adjudicator. (Color figure online)

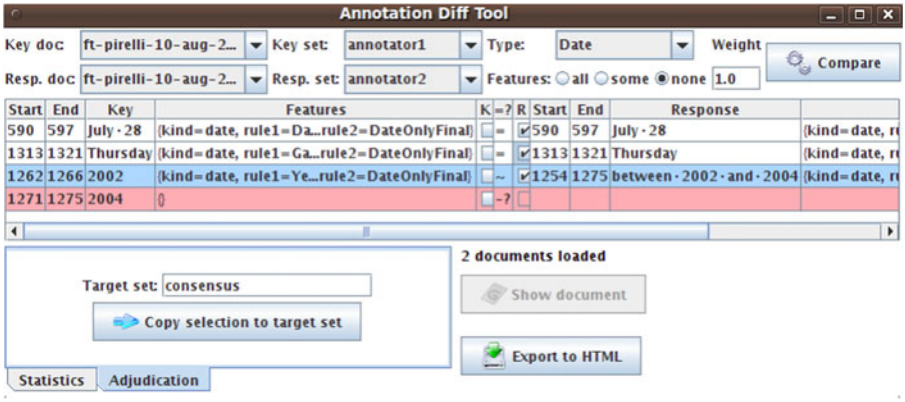

Fig. 4 The Annotation Diff user interface, showing side by side comparison of the annotations on which annotators differ

4.3.3 Project management interfaces

Apart from adjudication, project managers are responsible for defining annotation guidelines and schemas. They choose the relevant automatic services with which to pre- or post-process the data, benchmark annotator performance and monitor the

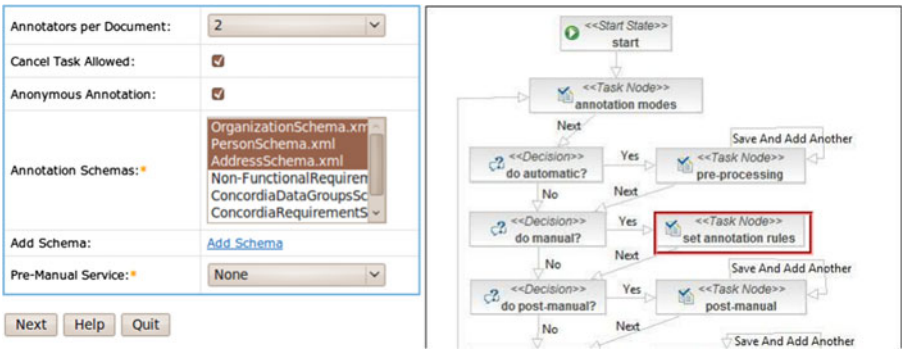

Fig. 5 Workflow Wizard, showing settings on the *left* and workflow design on the *right*

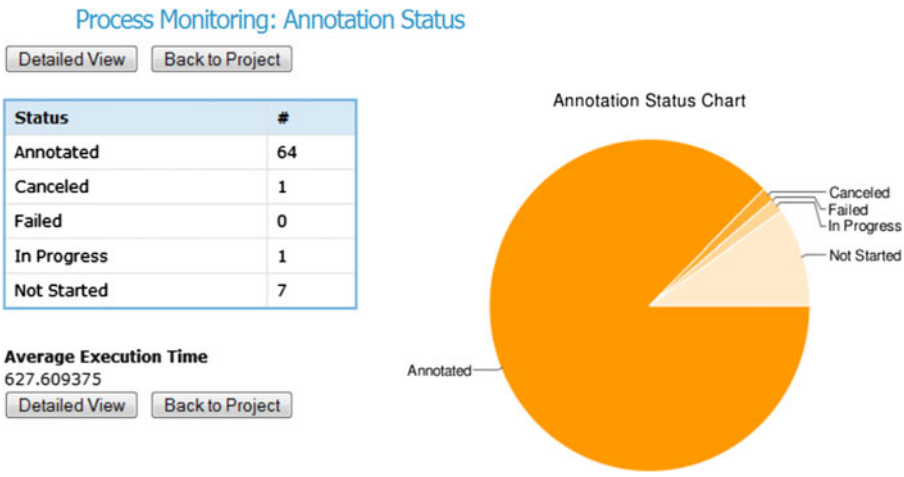

Fig. 6 The management user interface, showing the process monitoring screen, on which project progress is outlined on the *left* in terms of numbers of documents completed etc. and on the *right* in the form of a pie chart

project progress. Project managers define annotation workflows, manage annotators, and liaise with the system administrators.

The project management web UI provides the front-end to the executive layer (see Sect. 4.2). In a nutshell, managers upload documents and corpora, define the annotation schemas, specifying legal annotation types and legal attributes, choose and configure the workflows and execute them on a chosen corpus. Workflows may be as simple as passing the documents to *n* human annotators, or more complex, for example, preprocess the documents to produce automatic annotations, pass each document to three annotators and then adjudicate the differences. The workflow wizard facilitates this step, as shown in Fig. 5. The management console also provides project monitoring facilities, e.g. number of annotated documents, number in progress, and yet to be completed, as shown in Fig. 6. Per annotator statistics are

also available—time spent per document, overall time worked, average IAA, as well as per document statistics.

4.3.4 Administration user interface

Administrators can create, delete and suspend accounts, and can also use the GATE Teamware Bulk Upload feature to quickly add new user accounts from an Excel worksheet. Administrators can monitor processes and tasks that are created by GATE Teamware when projects are run.

5 Example corpus annotation projects

GATE Teamware has been used in practice in several corpus annotation projects of varying complexity and size, both by our group and by others. As mentioned in the introduction, Meurs et al. (2011) have applied GATE Teamware in the context of curating and extracting information from biomedical research papers to compile a database of fungal enzymes for use in biofuels. They used GATE Teamware to facilitate an existing manual-only effort in this domain, providing an opportunity to directly contrast the efficiency of the work with and without GATE Teamware. GATE was used to pre-annotate the papers, which were then manually corrected using GATE Teamware. A range of subject experts were used, demonstrating that GATE Teamware is usable by non-technologists with varied backgrounds. Their task comprised two stages. Firstly, papers were chosen for inclusion—a task previously requiring 2–3 min per paper. Secondly, papers were annotated—a task which previously required 30–45 min per paper. As a result of providing semantic and task support using GATE Teamware, times were reduced such that paper selection required only 1–2 min and paper annotation required only 20–30 min. Inter-annotator agreement is 80 %, when fixing up the automatically pre-annotated documents.

In this section, we present in more detail two other case studies, demonstrating the use of GATE Teamware. The target domains are business intelligence and bio-informatics, the latter being an ongoing collaborative project, annotating medical records.

5.1 Business intelligence evaluation

GATE Teamware has been applied in a commercial context, where a company had two teams of around 5 annotators (one in China and one in the Philippines). The annotation projects are being defined and overseen by managers in the USA, who also carry out adjudication. They have found that the standard double-annotated agreement-based approach is a good foundation for their commercial needs (e.g., in the early stages of the project and continuously for gold standard production), while they also used very simple workflows where the results of automatic services are being corrected by human annotators. They annotated over 1,400 documents, many

of which according to multiple schemas and annotation guidelines. For instance, 400 patent documents were doubly annotated both with measurements and bio-informatics entities, and then curated and adjudicated to create a gold standard. Their diverse user needs and practical experience with remote annotator teams exposed several issues to be addressed in future work:

- Different sets of annotators often work on the same corpus simultaneously, as part of separate projects, so that each team can specialise in a small number of annotation types. This requires support for merging the results of these separate projects into one consistent corpus, which is currently not achieved easily within the GATE Teamware environment, but is supported as a post-processing step in the GATE NLP development environment (Cunningham et al. 2011b).
- The annotator UI needs to be highly responsive to maximise the time annotators spend actually working on the documents. Consequently the data storage layer and the workflow need to minimise further network traffic, e.g., allow access to document-level metadata without also loading the entire document content.
- Execution speed of the annotation workflows needs to be optimised further, e.g., by avoiding unnecessary network traffic generated by temporary results being saved to the data store.
- Annotation of relations as well as manual annotation with medium- to large-size ontologies are required in many projects and the corresponding UIs need to support faster annotation.

5.2 Annotating biomedical texts

As part of the KHRESMOI biomedical research project,[9] GATE Teamware has been put to use in annotating two types of resources, in an ongoing effort. Firstly, radiology image captions are being annotated with references to anatomical parts and references to diseases, by a distributed team of annotators. Entities are cross-referenced with the UMLS[10] meta-thesaurus to ensure correctness. Secondly, other resources are also annotated, including Medline abstracts, gene home reference webpages and resource websites on diseases such as diabetes. These have been crawled and downloaded to create an informational resource with which patient records may be cross-referenced.

5.2.1 Annotators and methodology

The 15 annotators are medically trained and located in the Philippines. Initially, 30 annotators were involved, but later, their number dropped to 15. Of these, three are managers. All annotators speak excellent English. One of the authors spent a day training and observing the annotators in using GATE Teamware; since then, problems have been discussed via telephone or email.

[9] http://www.khresmoi.eu/.

[10] http://www.nlm.nih.gov/research/umls/.

Table 2 Annotator performance on Gene home reference webpages, showing consensus and inter-annotator agreement, micro- and macro-averaged, for four different task variants

Annotation	Consensus macro-avg.	Consensus micro-avg.	IAA macro-avg.	IAA micro-avg.
Anatomy, no features	0.91	0.78	0.87	0.69
Anatomy, UMLS ID	0.90	0.69	0.85	0.61
Disease, no features	0.91	0.82	0.90	0.77
Disease, UMLS ID	0.88	0.75	0.88	0.70

Each text is annotated by two annotators. A manager then curates the work. Where the two annotators agree, the manager simply accepts their judgement. Where they disagree, the manager will adjudicate. They also look for any missed annotations.

5.2.2 Results

Five annotation rounds have been completed, with a sixth ongoing. In the first round 14 users annotated anatomical parts in 250 image captions, with macro-average IAA was 0.62. After some training, the second round improved IAA to 0.84. Subsequent rounds included another 250 image captions and also annotations with diseases and medical issues.

The project also annotated 250 longer documents (gene home reference webpages) and tested the inclusion of more complex information within the text annotations. In particular, the task was to annotate mentions of diseases and anatomy concepts and assign the corresponding concept identifiers from the UMLS meta-thesaurus. Table 2 shows the details.

5.2.3 Improving quality control tools

This annotation project provided the impetus to develop one more quality control tool, to support the use of GATE Teamware in larger projects. The QA Summariser (Fig. 7) generates a summary of agreements among annotators. It does this by pairing individual annotators involved in the annotation task. It also compares the annotations of each individual annotator against those in the consensus set.

The tool generates an index.html file in the output folder. This HTML file contains a table that summarises the agreement statistics. Both the first row and the first column contain names of annotators who were involved in the annotation task. For each pair of annotators who did the annotations together on at least one document, both the micro and macro averages are produced.

The last two columns in each row give average macro and micro agreements of the respective annotator with all the other annotators with whom they annotated documents together.

Agreement scores are colour-coded. The colour green is used for a cell background to indicate full agreement (1.0). The background colour becomes lighter

Summary of IAA Results

AnnotationTypes: Anatomy;
Features:
Measure: F1_AVERAGE

Author Names	consensus		hsb		ith		jlb		mmb		mmm		mpr		osv		rbm		rea		rip		Averages	
	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro
consensus			0.95	0.85	0.93	0.86	0.95	0.77	0.90	0.71	0.84	0.62	0.98	0.92	0.92	0.85	0.85	0.67	0.95	0.81	0.86	0.71	0.91	0.78
			document		document		document		document		document		document		document		document		document		document			
hsb	0.95	0.85			0.92	0.69	0.95	0.78	0.91	0.72	0.77	0.62	0.87	0.68	0.92	0.80	0.67	0.33	0.96	0.84	0.93	0.73	0.88	0.70
	document				document		document		document		document		document		document		document		document		document			
ith	0.93	0.86	0.92	0.69			0.91	0.67	0.80	0.45	0.73	0.57	0.96	0.78	0.87	0.80	0.78	0.65	0.88	0.64	0.80	0.82	0.86	0.69
	document		document				document		document		document		document		document		document		document		document			
jlb	0.95	0.77	0.95	0.78	0.91	0.67			0.89	0.53	0.87	0.62	0.94	0.66	0.96	0.80	0.89	0.77	0.94	0.74	0.77	0.52	0.91	0.69
	document		document		document				document		document		document		document		document		document		document			
mmb	0.90	0.71	0.91	0.72	0.80	0.45	0.89	0.53			0.96	0.67	0.91	0.67	0.81	0.56	1.00	1.00	0.89	0.82	0.73	0.81	0.88	0.69
	document		document		document		document				document		document		document		document		document		document			
mmmisa	0.84	0.62	0.77	0.62	0.73	0.57	0.87	0.62	0.96	0.67			0.73	0.38	0.89	0.69	0.74	0.59	0.92	0.83	0.98	0.86	0.84	0.64
	document		document		document		document		document				document		document		document		document		document			
mpr	0.98	0.92	0.87	0.68	0.96	0.78	0.94	0.66	0.91	0.67	0.73	0.38			0.89	0.85	0.79	0.59	0.93	0.80	0.86	0.74	0.89	0.71
	document		document		document		document		document		document				document		document		document		document			
osv	0.92	0.85	0.92	0.80	0.87	0.80	0.96	0.80	0.81	0.56	0.89	0.69	0.89	0.85			0.88	0.74	0.93	0.81	0.83	0.44	0.89	0.73
	document		document		document		document		document		document		document				document		document		document			
rbm	0.85	0.67	0.67	0.33	0.78	0.65	0.89	0.77	1.00	1.00	0.74	0.59	0.79	0.59	0.88	0.74			0.67	0.00	0.89	0.67	0.82	0.60
	document		document		document		document		document		document		document		document				document		document			
rea	0.95	0.81	0.96	0.84	0.88	0.64	0.94	0.74	0.89	0.82	0.92	0.83	0.93	0.80	0.93	0.81	0.67	0.00			0.97	0.87	0.90	0.72
	document		document		document		document		document		document		document		document		document				document			
rip	0.86	0.71	0.93	0.73	0.80	0.82	0.77	0.52	0.73	0.81	0.96	0.86	0.86	0.74	0.83	0.44	0.89	0.67	0.97	0.87			0.86	0.72
	document		document		document		document		document		document		document		document		document		document					

Avg. consensus macro avg: 0.91
Avg. consensus micro avg: 0.78
Avg. IAA macro avg: 0.87
Avg. IAA micro avg: 0.69

Fig. 7 Quality assurance summariser for Teamware, showing for each annotator their micro and macro consensus with each other annotator. *Colour* is used to indicate the extent of consensus and links to per document results are provided. (Color figure online)

as the agreement reduces towards 0.5. At 0.5 agreement, the background colour of a cell is fully white. From 0.5 downwards, the colour red is used and as the agreement reduces further, the colour becomes darker with dark red at 0.0 agreement. Use of such colour coding makes it easy for a manager to get an overview of annotator performance and identify problematic annotators.

For each pair of annotators, the summary table provides a link (with a caption document) to another HTML document that summarises annotations of the two respective annotators on a per document basis. The details include number of annotations they agreed/disagreed, the f-measure score, and table-based comparison of the actual annotations.

6 Teamware as a cloud service

From an implementational perspective, GATE Teamware uses a 3-tier service-based architecture for distributed collaborative annotation, driven by a centralised workflow engine. The architecture is appropriate to the task and performs effectively, but deployment and administration of the tool is complex and error-prone. In order to address this problem, we have developed a cloud-based deployment where a standard virtual machine image is supplied.[11] Instead of needing skilled administrator time, the process is fully automatic and results in a running GATE Teamware instance that can be turned on and off as required.

In addition, the GATECloud.net integration makes it straightforward to select a sample from all automatically annotated documents on the cloud and channel these

[11] Cloud-based GATE Teamware virtual machines are available at http://gatecloud.net/.

into a cloud-based GATE Teamware instance, where human annotators correct them in order to create a gold standard corpus.

Since typically, manual corpus annotation is an activity undertaken for relatively short periods of time (days, weeks and rarely one or two months) by teams of people, such a service-based approach cuts costs and offers flexibility with respect to intermittent usage.

7 Conclusions and future work

This paper described GATE Teamware—a multi-role, web-based annotation environment, which supports customised text annotation workflows and provides methodological and tool support to the different actors involved in the process. It fully supports the complete lifecycle of annotation projects, as defined by Müller and Strube (2006): data preparation, schema definition, performing annotation, agreement and adjudication, and use (i.e. corpus query, API access).

GATE Teamware, and in particular its cloud-based on-demand deployment, has a number of target user categories. Firstly, researchers working in smaller groups and/or on small NLP projects, who need text annotation for a short period only and typically do not have the resources to install separate, large annotation tools or develop new ones. Secondly, companies can use it in a secure environment, in order to develop their own corpora or to offer corpus annotation services to others. Thirdly, PhD students can use the GATE Teamware service to create training data, if such does not exist, either by themselves or by involving other student volunteers. Lastly, companies working with sensitive data, such as patient records, have requested virtual machine images with pre-configured GATE Teamware, which can be used in-house without any installation effort.

Evaluation with distributed annotator teams working on a range of corpus annotation projects has shown the flexibility and utility of the system. The teams have involved both less experienced annotators (e.g. biology students), as well as more advanced domain experts, who were trained more extensively in using advanced features, such as regular expressions, in order to improve their productivity.

The corpus annotation projects also uncovered some limitations that need to be addressed in future work. More specifically, GATE Teamware does not currently provide good support for relation and co-reference chain annotation. However, GATE Developer offers these interface components, so experienced software developers, familiar with the GATE APIs, will be able to extend GATE Teamware with these plugins, since they are fully compatible. The underlying stand-off annotation model would not need any modifications.

Another area of future work is in automating the merging of results of separate annotation projects into one consistent corpus, improving the speed and responsiveness of the annotation interfaces and automatic pre-annotation services, as well as improving user interface ergonomics.

Acknowledgments The authors wish to thank Milan Agatonovic and Thomas Heitz who worked on GATE Teamware whilst being at Sheffield. We also wish to thank Matthew Petrillo, Jessica Baycroft, and Danica Damljanovic for running some of the distributed annotation experiments and allowing us to report the results here. The paper was also improved greatly thanks to the helpful suggestions of the three anonymous reviewers. The first author is being supported by funding from the Engineering and Physical Sciences Research Council (grant EP/I004327/1).

References

ACE. (2004). *Annotation guidelines for event detection and characterization (EDC)*. Available at http://www.ldc.upenn.edu/Projects/ACE/.

Agatonovic, M., Aswani, N., Bontcheva, K., Cunningham, H., Heitz, T., Li, Y., et al. (2008). Large-scale, parallel automatic patent annotation. In *Proceedings of the 1st ACM workshop on patent information retrieval (PaIR '08)*, pp. 1–8.

Berck, P., & Russel, A. (2006). Annex a web-based framework for exploiting annotated media resources. In *Proceedings of the fifth international conference on language resources and evaluation*.

Brugman, H., & Russel, A. (2004). Annotating multi-media/multi-modal resources with elan. In *Proceedings of the fourth international conference on language resources and evaluation*.

Brugman, H., Crasborn, O., & Russel, A. (2004). Collaborative annotation of sign language data with peer-to-peer technology. In *Proceedings of LREC*.

Carletta, J., Evert, S., Heid, U., & Kilgour, J. (2005). The nite xml toolkit: Data model and query language. *Language Resources and Evaluation, 39*(4), 313–334.

Cunningham, H., Maynard, D., Bontcheva, K., & Tablan, V. (2002). Gate: An architecture for development of robust hlt applications. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 168–175.

Cunningham, H., Fuhr, N., & Stein, B. (Eds). (2011a). *Challenges in document mining—report from Dagstuhl seminar 11171*. Dagstuhl reports. Dagstuhl, Germany: Leibniz-Zerntrum für Informatik.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., et al. (2011b). *Text processing with GATE (version 6)*. The University of Sheffield. http://tinyurl.com/gatebook.

Day, D., Aberdeen, J., Hirschman, L., Kozierok, R., Robinson, P., & Vilain, M. (1997). Mixed-initiative development of language processing systems. In *Proceedings of the 5th conference on applied natural language processing (ANLP-97)*.

Day, D., McHenry, C., Kozierok, R., & Riek, L. (2004) Callisto: A configurable annotation workbench. In *International conference on language resources and evaluation*.

Ferrucci, D., & Lally, A. (2004). UIMA: An architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering, 10*(3–4), 327–348.

Hovy, E. (2010). Annotation. In *Tutorial Abstracts of ACL*.

Hovy, E., Marcus, M. P., Palmer, M., Ramshaw, L. A., & Weischedel, R. M. (2006). Ontonotes: The 90 % solution. In *Proceedings of HLT-NAACL*.

Ide N., & Romary, L. (2004) Standards for language resources. *Natural Language Engineering*, 10, 211–225.

Ide, N., & Suderman, K. (2005). Integrating linguistic resources: The american national corpus model. In *Proceedings of human language technology conference/conference on empirical methods in natural language processing HLT/EMNLP 2005*, Vancouver, B.C., Canada.

Ide, N., Bonhomme, P., & Romary, L. (2000) XCES: An XML-based standard for Linguistic Corpora. In *Proceedings of the second international conference on language resources and evaluation (LREC 2000), 30 May—2 Jun 2000*, Athens, Greece, pp. 825–830. http://www.lrec-conf.org/proceedings/lrec2000/pdf/172.pdf.

Kiryakov, A. (2006) OWLIM: Balancing between scalable repository and light-weight reasoner. In *Proceedings of the 15th international world wide web conference (WWW2006), 23–26 May 2010*, Edinburgh, Scotland. http://www.ontotext.com/sites/default/files/publications/Kiryakov_OWLIM_www2006.pdf.

Maeda, K., & Strassel, S. (2004). Annotation tools for large-scale corpus development: Using AGTK at the linguistic data consortium. In *Proceedings of 4th language resources and evaluation conference (LREC'2004)*.

Maeda, K., Lee, H., Medero, S., Medero, J., Parker, R., & Strassel, S. (2008). Annotation tool development for large-scale corpus creation projects at the linguistic data consortium. In *Proceedings of the sixth international language resources and evaluation (LREC'08)*.

Maynard, D., Funk, A., & Peters, W. (2009). SPRAT: A tool for automatic semantic pattern-based ontology population. In *International conference for digital libraries and the semantic web*, Trento, Italy.

Meurs, M. J., Murphy, C., Naderi, N., Morgenstern, I., Cantu, C., & Semarjit, S., et al. (2011). Towards evaluating the impact of semantic support for curating the fungus scientific literature. In *The 3rd Canadian semantic web symposium (CSWS2011)*, Vancouver, B.C., Canada.

Müller, C., & Strube, M. (2006). Multi-level annotation of linguistic data with MMAX2. In S. Braun, K. Kohn, & J. Mukherjee (Eds.) *Corpus technology and language pedagogy: New resources, new tools, new methods* (pp. 197–214). Germany: Peter Lang, Frankfurt a.M.

Ogren, P. (2006). Knowtator: A Protege plug-in for annotated corpus construction. In *HLT-NAACL—Demos*.

Poesio, M., Diewald, N., Stührenberg, M., Chamberlain, J., Jettka D., Goecke, D., et al. (2012). Markup infrastructure for the anaphoric bank: Supporting web collaboration. In A. Mehler, K. U. Kühnberger, H. Lobin, H. Lüngen, A. Storrer, & A. Witt (Eds.), *Modeling, learning, and processing of text technological data structures, studies in computational intelligence* (Vol. 370, pp. 175–195). Berlin, Heidelberg: Springer.

Tablan, V., Ursu, C., Bontcheva, K., Cunningham, H., Maynard, D., Hamza, O., et al. (2002). A unicode-based environment for creation and use of language resources. In *Proceedings of the 3rd language resources and evaluation conference*.

Tablan, V., Roberts, I., Cunningham, H., & Bontcheva, K. (2013). Gatecloud.net: a platform for large-scale, open-source text processing on the cloud. *Philosophical Transactions of the Royal Society A, 371*(1983). doi:10.1098/rsta.2012.0071.

Wiebe, J., Wilson, T., & Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation, 39*(2–3), 165–210.